APSS Assignment 2

SID: 490423699

Time taken to complete assignment: $\sim 10\ hrs$

Question 1:

(i) Derivation of the acceleration equation:

i)

$$\left(\frac{\dot a}{a}\right)^2 = \frac{8\pi G}{3c^2}\varepsilon - \frac{Kc^2}{R_0^2 a^2}$$

Multiply both sides by a^2

$$\Rightarrow \dot a^2 = \frac{8\pi G}{3c^2}\varepsilon a^2 - \frac{Kc^2}{R_0^2}$$

Now take the time derivative:

$$\frac{d}{dt}\left(\dot a^2\right) = \frac{d}{dt}\left[\frac{8\pi G}{3c^2}\varepsilon a^2 - \frac{Kc^2}{R_0^2}\right]$$

$$2\dot a \ddot a = \frac{8\pi G}{3c^2}\frac{d}{dt}\left(\varepsilon a^2\right)$$

$$= \frac{8\pi G}{3c^2}\left(\dot\varepsilon a^2 + 2\varepsilon a \dot a\right)$$

$$2\dot a \ddot a = \frac{8\pi G}{3c^2}\left(\dot\varepsilon a^2 + 2\varepsilon a\dot a\right)$$

Now, divide through by $2a\dot a$:

$$① \quad \frac{\ddot a}{a} = \frac{4\pi G}{3c^2}\left(\frac{\dot\varepsilon a}{\dot a} + 2\varepsilon\right)$$

Using fluid equation, $\frac{\dot\varepsilon a}{\dot a} = -3(\varepsilon + P)$

$\Rightarrow$ Subs into ①

We have:

$$\frac{\ddot a}{a} = \frac{4\pi G}{3c^2}\left(-3\varepsilon - 3P + 2\varepsilon\right)$$

$$= \frac{4\pi G}{3c^2}\left(-3P - \varepsilon\right)$$

$$\boxed{\frac{\ddot a}{a} = -\frac{4\pi G}{3c^2}\left(3P + \varepsilon\right)}$$ Acceleration Eqn.

(ii) Variables across the three equations and their meanings, as well as assumptions made when solving these equations.

Friedmann equation:

- ϵ is the energy density of the universe (from all possible contributions), which also varies as a function of time.
- κ is a curvature parameter (index) in the FRW metric which describes the spatial curvature of the universe. It can take one of three values: (+1) which denotes a positive curvature, (-1) which denotes a negative curvature, or 0 which implies a flat universe.
  - R_0 is the present value of the curvature radius.
- a is scale factor which captures the size of the universe at a given time, relative to its current size (where a is arbitrarily chosen to be 1 at present, by convention) – since the scale factor evolves with time, we denote it as $a(t)$.
- $\dot{a}$ is the time derivative of the scale factor, for negative values, this implies a contracting universe and for positive values, this indicates an expanding universe.

Fluid Equation – describes the evolution of energy density in an expanding universe:

- $\dot{\epsilon}$ is the time derivative of the energy density of the universe (from all possible contributions).
- a is scale factor which captures the size of the universe at a given time, relative to its current size (where a is arbitrarily chosen to be 1 at present, by convention) – since the scale factor evolves with time, we denote it as $a(t)$.
- $\dot{a}$ is the time derivative of the scale factor, for negative values, this implies a contracting universe and for positive values, this indicates an expanding universe.
- ϵ is the energy density of the universe (from all possible contributions), which also varies as a function of time.
- P is the pressure of a CMB photon gas.

Acceleration Equation – describes how the acceleration of the expansion of the universe is influenced by its energy content.

- $\ddot{a}$ is the second time derivative of the scale factor, which corresponds to the acceleration of the expansion of the universe.
- a is scale factor which captures the size of the universe at a given time, relative to its current size (where a is arbitrarily chosen to be 1 at present, by convention) – since the scale factor evolves with time, we denote it as $a(t)$.
- ϵ is the energy density of the universe (from all possible contributions), which also varies as a function of time.
- P is the pressure of a CMB photon gas.

Assumptions made solving these equations:

- Assume that the universe is homogenous and isotropic such that it can be modelled as a perfect fluid.
- Assume adiabatic expansion of a homogeneous universe i.e., treat the universe as a closed system which has no external surrounding to which heat energy can be transferred (heat flow dQ = 0). This allows us to obtain the fluid equation from the first law of thermodynamics.

Question 2

(i)

We have:

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3c^2}\varepsilon(t) - \frac{\kappa c^2}{R_0^2 a^2}$$

Assuming flat universe ($\kappa=0$) and radiation dominated ($\omega = 1/3$)

$P = \omega\varepsilon$

$P = \frac{\varepsilon}{3}$ → Fluid eqn: $\dot{\varepsilon} + 3\frac{\dot{a}}{a}(\varepsilon + P) = 0$

$$\dot{\varepsilon} + 3\frac{\dot{a}}{a}(\varepsilon + \varepsilon/3) = 0$$

$$\Rightarrow \dot{\varepsilon} + 4\varepsilon\frac{\dot{a}}{a} = 0$$

$$\varepsilon \propto 1/a^4$$

Now,

$$\frac{\dot{a}^2}{a^2} \propto \frac{8\pi G}{3c^2}\frac{1}{a^4}$$

$$\dot{a}^2 a^2 \propto \frac{8\pi G}{3c^2}$$

$\dot{a}a \propto \sqrt{\frac{8\pi G}{3c^2}}$ → Subs $\dot{a} = \frac{da}{dt}$

$$\int a\,da \propto \int\sqrt{\frac{8\pi G}{3c^2}}\,dt$$

$$\frac{a^2}{2} \propto \sqrt{\frac{8\pi G}{3c^2}}\,t$$

$$a^2 \propto \left(\frac{32\pi G}{3c^2}\right)^{1/2} t$$

$$a \propto \left(\frac{32\pi G}{3c^2}\right)^{1/4} t^{1/2}$$

$$\therefore a(t) \propto t^{1/2}$$

(2) Matter dominated ($\omega=0$) & $\varepsilon \propto a^{-3}$

$$\left(\frac{\dot{a}}{a}\right)^2 \propto \frac{8\pi G}{3c^2}\frac{1}{a^3}$$

$$\frac{\dot{a}^2}{a^2} \propto \frac{8\pi G}{3c^2}\frac{1}{a^3}$$

$$\dot{a}^2 a \propto \frac{8\pi G}{3c^2}$$

$$\dot{a}^2 \propto \frac{8\pi G}{3c^2}\frac{1}{a}$$

$$\dot{a} \propto \sqrt{\frac{8\pi G}{3c^2}\frac{1}{a}}$$

$$\frac{da}{dt}a^{1/2} \propto \sqrt{\frac{8\pi G}{3c^2}}$$

$$\int a^{1/2}\,da \propto \int\sqrt{\frac{8\pi G}{3c^2}}\,dt$$

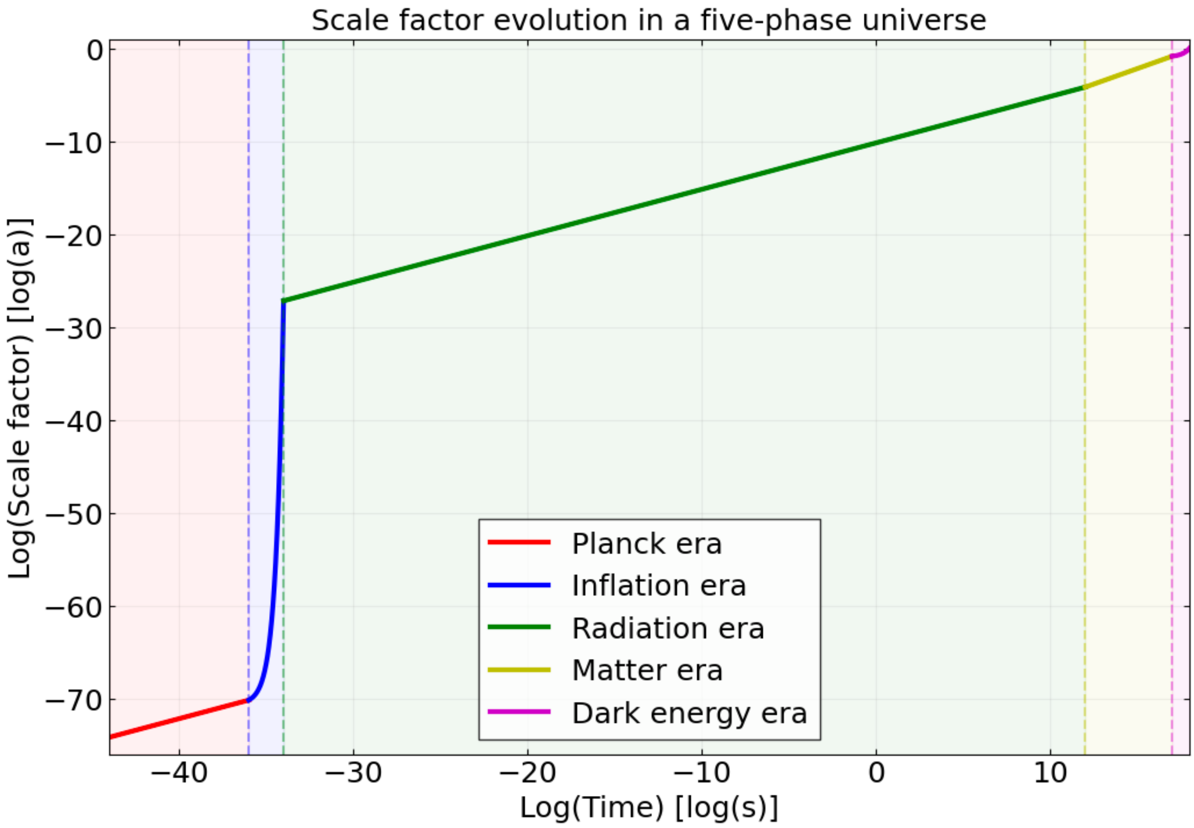
$$\frac{2}{3}a^{3/2} \propto \sqrt{\frac{8\pi G}{3c^2}}\,t$$

$$a^{3/2} \propto \frac{3}{2}\sqrt{\frac{8\pi G}{3c^2}}\,t$$

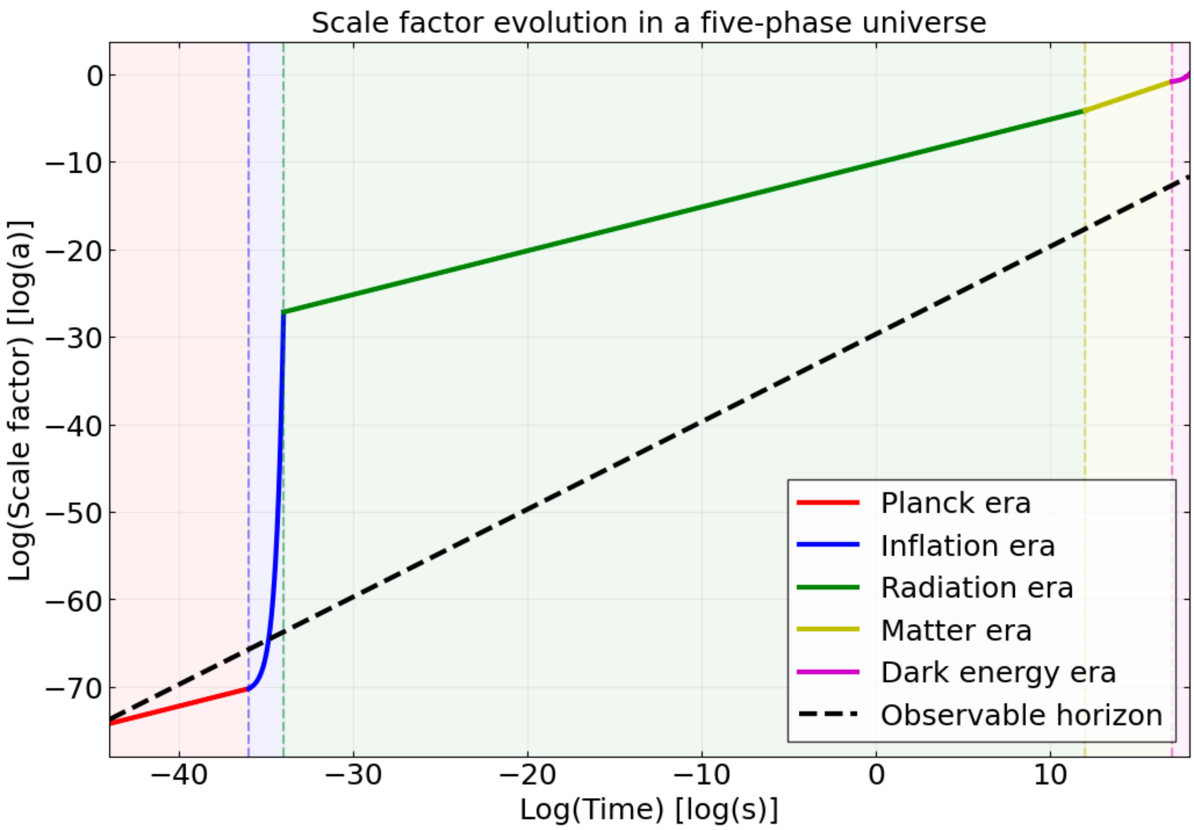
$$a(t) \propto t^{2/3}$$

(ii) Plot of scale factor through five phases of evolution:

Using the values and times (for each era) quoted in the lectures, I was unable to generate a curve in which log(a) was 0 at $t = 10^{18}s$, so I offset the curve slightly to enforce this condition. However, the original trends are preserved. The un-modified version of the plot is also available in the Jupyter notebook.



(iii) Same plot with the horizon (passage of light) overlayed:



At the current time where $t = 10^{18} s$, there is a discrepancy between the observable horizon and the size of the universe according to this five-phase universe model. This implies that Universe is much larger than what can be currently observed, as light from the early universe has not had enough time to travel to us.

Question 3:

(i) Describe how Guth's inflation model solved at least three fundamental cosmological problems that plagued cosmology before 1981.

Guth's model of inflation addressed three fundamental problems which, prior to its proposal, had posed a challenge to field of cosmology: (1) the flatness problem, (2) the horizon problem and (3), the magnetic monopole problem.

One of the most intriguing observations that plagued cosmology before 1981 was that the spatial distribution of density fluctuations, as inferred from CMB temperature measurements, is isotropic and homogeneous in all directions, despite remote regions of the universe being separated by distances too large for light to enable causal contact (and thereby equilibrate) – this became known as the 'Horizon problem'. In Guth's model, prior to inflation, the universe was sufficiently small that these distant regions were once close enough to be well within each other's horizon of causal contact, and therefore were able to interact and exchange information to establish a state of thermal equilibrium. During the inflationary period, these regions were stretched further apart, however the established equilibrium was preserved and is what we measure today. Thus, Guth's model addresses the Horizon problem by proposing that once causally connected regions (in a state of thermal equilibrium) became spatially separated and seemingly disconnected due to cosmic inflation.

Another question that had plagued cosmologists was the so called 'flatness' problem: measurements from the CMB have determined the geometry of the universe to be nearly flat. A more subtle point lies in the realisation that to achieve this flatness we observe today, this requires that the density of the universe be precisely 'fine-tuned' near some critical density. Even slight deviations from this critical value would result in curvature that becomes amplified over time, resulting in different conditions in the geometry of the resulting universe. This is somewhat reminiscent of chaos theory, where a small perturbation in a system's (universe's) initial conditions rapidly diverge into significantly distinct outcomes (spatial geometries). Guth's model of inflation addresses this sensitive dependence on initial conditions and removes the need for cosmological 'fine-tuning' by proposing that during the inflationary phase, any prior perturbations in curvature were diluted (smoothed) by the rapid expansion of the universe, resulting in a spatially flat geometry.

The monopole problem is one that stems from attempting to reconcile the Big Bang with the so called "Grand Unified Theory" (GUT) which implies that under high energy conditions, such as those during the very early universe, the electromagnetic, strong, and weak nuclear forces were unified into a single force. One prediction of GUT is that the universe underwent a phase transition as the temperature of the universe dropped below the GUT temperature ($T_{GUT} \sim 10^{28} K$) (Ryden, 2003), that is, the temperature at which the strong and electroweak forces decouple and begin to behave differently. During this phase transition, GUT predicts that because of spontaneous symmetry breaking, point-like defects which act as magnetic monopoles should have been produced in large numbers, and that the energy density of these monopoles would have dominated the energy density of the early universe. Furthermore, these magnetic monopoles should persist to the present day. Nevertheless, there is no strong evidence for the existence of magnetic monopoles in the current universe. Guth's inflationary theory suggests that if magnetic monopoles had indeed existed before or during the epoch of inflation, then post-inflation, the number density would have been diluted by many orders of magnitude such that the probability of detection in the present day would be extremely small. As an example, Ryden (Ryden, 2003) states that if inflation began at the GUT time with a magnetic monopole density of $n_M(t_{GUT}) \approx 10^{82} m^{-3}$, then after 100 e-foldings of inflation, and the additional expansion since, the number density of monopoles in the present universe would be $n_M(t_0) \approx 10^{-61} Mpc^{-3}$, an astronomically small number density. Therefore, if magnetic monopoles exist, the likelihood of detecting one is incredibly low, according to Guth's inflation model.

(ii) Why does an "empty" universe filled only with a vacuum (no particles, no dark sector) still expand?

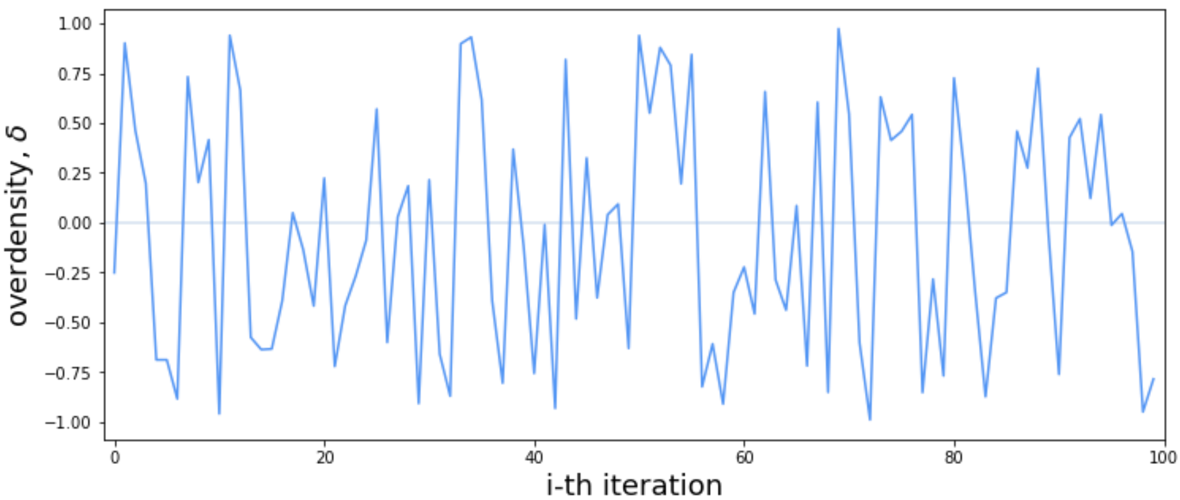
A vacuum is never truly 'empty' in the proper sense as this would violate Heisenberg's uncertainty principle:

$$\Delta E \, \Delta t \, \geq \frac{1}{2} \hbar$$

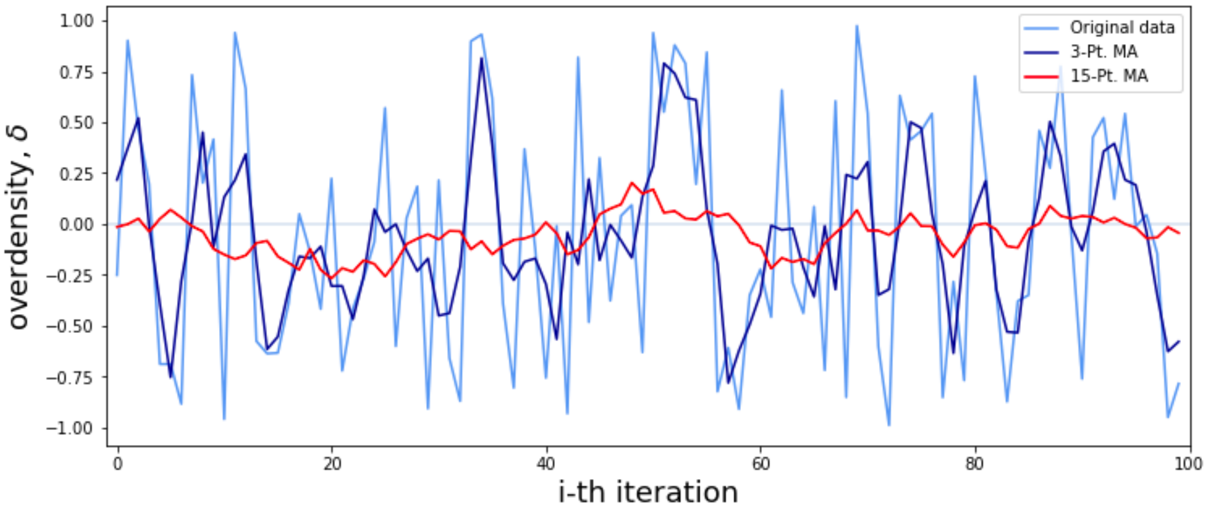
Here, an empty universe with zero energy content would imply $\Delta E = 0$ (and thereby violate Heisenberg's uncertainty principle). Rather, a direct consequence of Heisenberg's uncertainty principle is that over small-time intervals, there can be large uncertainties in energy – such uncertainties manifest as vacuum fluctuations. An empty universe is permeated by such a vacuum with an associated energy, called the vacuum energy, that is inherent to the vacuum itself. This vacuum energy arises from these quantum fluctuations, arising from the creation and annihilation of particle-antiparticle pairs. Most importantly, the vacuum energy drives expansion by exerting a negative pressure.

Question 4:

(i) Random number sequence generated by sampling from a uniform distribution:



(ii) Original data with moving average of 3 points and 15 points overlayed:



(iii) Transforming from a Uniform to a normal (Gaussian) distribution:

While several algorithms and techniques exist for converting from a uniform distribution of random numbers to a Gaussian distribution, the Box-Muller transform is perhaps one of the most ubiquitous and computationally efficient. There are two formulations of the transformation, the basic formulation (cartesian) involves the following procedure (Box & Muller, 1958):

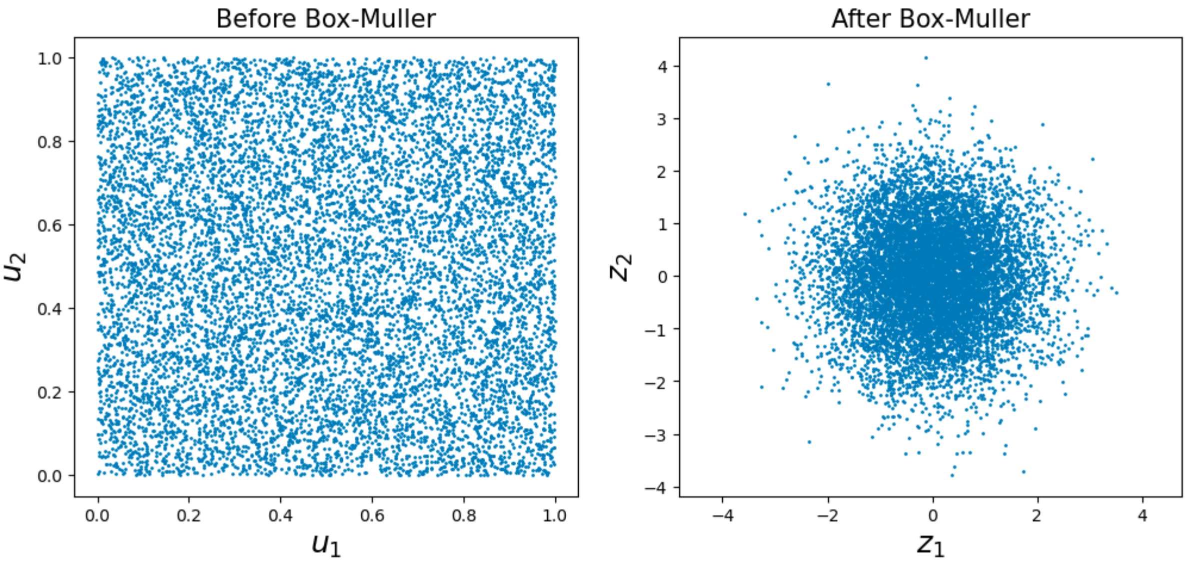
1. Generate two random numbers, U_1 and U_2 from a uniform distribution defined on the interval [0,1].
2. Apply the following transformations which convert U_1 and U_2 to Z_1 and Z_2:

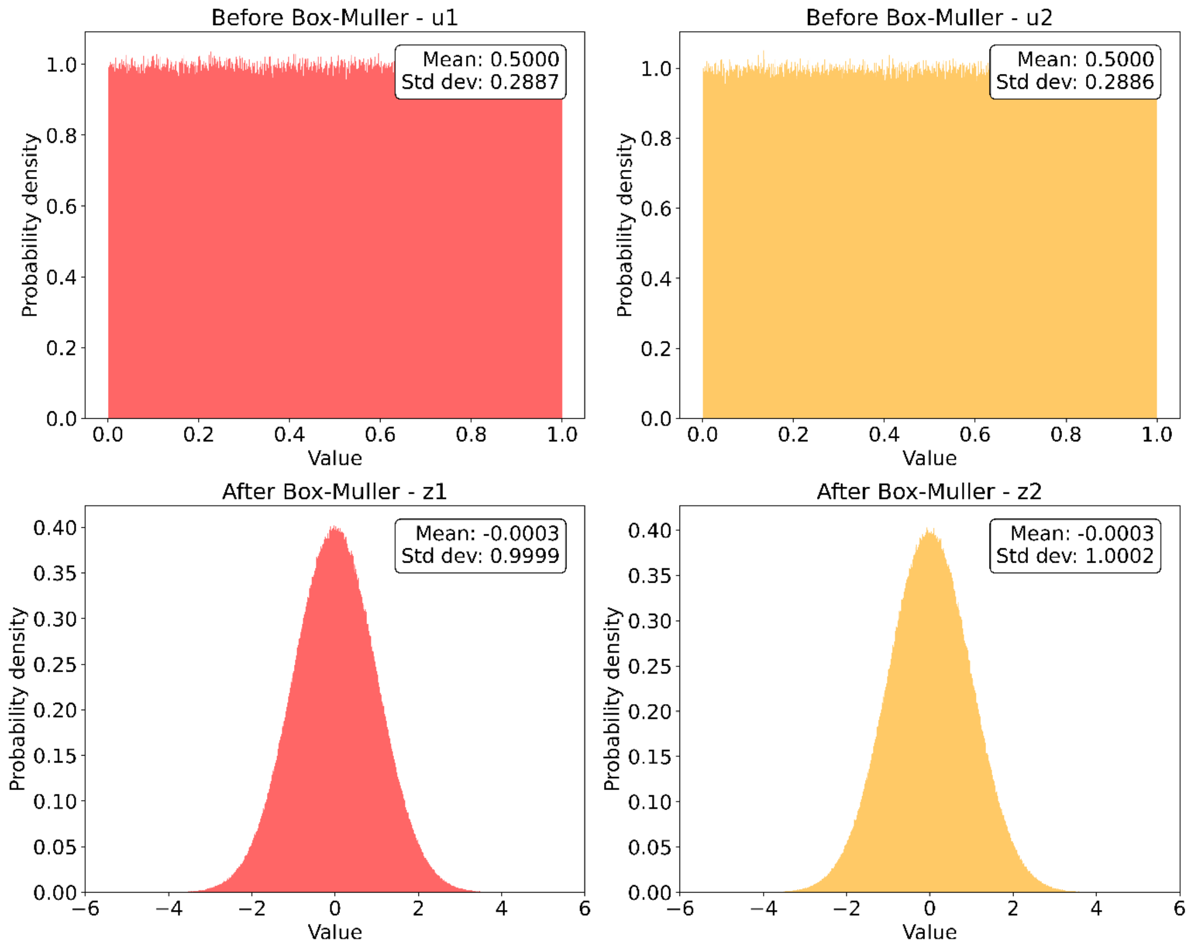
$$Z_1 = \sqrt{-2lnU_1}\cos(2\pi U_2)$$

$$Z_2 = \sqrt{-2lnU_1}\sin(2\pi U_2)$$

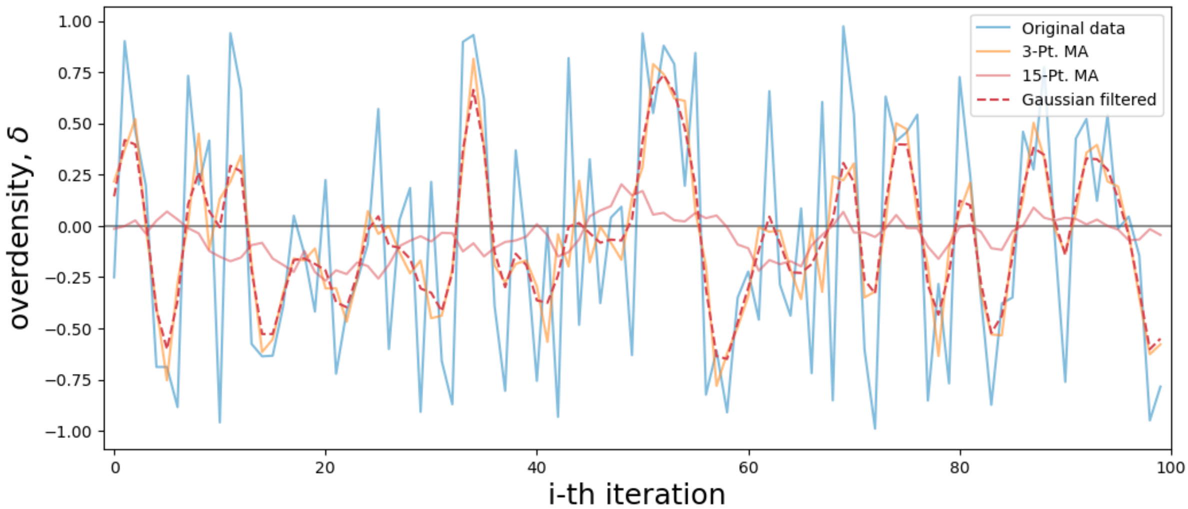
   Here, Z_1 and $Z_2$are independent random variables from a Gaussian distribution of mean 0 and variance 1 i.e., $Z_1, Z_2 \sim \mathcal{N}(0,1)$.
3. Repeat the transformation for as many samples as required to generate an entire normal distribution.

In the plots below, I demonstrate a basic implementation of the Box-Muller transform on 10 000 samples from a uniform distribution. I create a scatter plot whereby each random samples occupies a point in a 2D space. As expected, the points are uniformly distributed over the interval [0, 1] prior to the Box-Muller transform (on the left), while post-transformation, the corresponding scatter plot resembles that expected for a normal Gaussian distribution with mean 0 and variance 1 (as per the analysis in the attached Jupyter notebook). This is perhaps clearer in the attached histograms comparing the distributions before and after the Box-Muller transformation.





(iv) Original data with Gaussian convolution (smoothing) overlayed:



The result of the Gaussian convolution is different to that of the two moving averages due to the differences in the shapes of the underlying functions. For example, the top-hat filter, as the name implies, resembles the shape of a top hat with a constant value within the sliding window (implying equal weightings to the points under consideration), before abruptly dropping to zero at the edges. As a result, convolution with this function results in a signal that is 'sharper' and less smooth, particularly in the case of a 3-point moving average. By contrast, the Gaussian function is smooth and continuous, resulting in a signal with less abrupt transitions and is therefore smoother in general. As with the 3-point moving average, the filtered signal more closely resembles the shape of the original signal as the immediate neighbouring points for any given position i in the random number sequence has a higher weighting, and thus more influence on the smoothing, than other more distant ones.

Code Availability

Code is available as a Jupyter notebook (python) called 'APSS_Assignment2_2023.ipynb' at my personal GitHub repository: https://github.com/jmoo2880/honours-coursework as well as directly below.

References

Box, G. E. P., & Muller, M. E. (1958). A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics*, *29*(2), 610–611. https://doi.org/10.1214/aoms/1177706645

Ryden, B. Sue. (2003). *Introduction to cosmology*. Addison-Wesley.

Code

Q2 (ii) and (iii) Scale factor evolution plots

```
a_i = 1e-57
t_i = 1e-36  # beginning of inflationary period
t_f = 1e-34  # end of inflationary period
H_i = 1e36  # Hubble constant at the beginning of inflation in s^-1
H_0 = 2.2e-18  # Hubble constant today in s^-1
t_now = 1e18

t1 = np.linspace(1e-44, t_i, 1000)
a1 = a_i * (t1 / t_i)**(1 / 2)  # phase one - radiation dominated for t
< t_i

t2 = np.linspace(t_i, t_f, 1000)
a2_start = a1[-1]
a2 = a2_start * np.exp(H_i * (t2 - t_i))  # phase two - inflation

t3 = np.linspace(t_f, 1e12, 1000)
a3_start = a2[-1]
a3 = a3_start * (t3 / t_f)**(1 / 2)  # phase three - reheating and
radiation dominated

t4 = np.linspace(1e12, 1e17, 1000)
a4_start = a3[-1]
a4 = a4_start * (t4 / t3[-1])**(2 / 3)  # phase four - matter era

t5 = np.linspace(1e17, t_now, 1000)
a5_start = a4[-1]
a5 = a5_start * np.exp(H_0 * (t5 - t5[0]))  # phase five - dark energy
era

# observable horizon
t6 = np.linspace(1e-44, 1e18, 1000)
a6 = 3e8 * t6 * 1e-25 # multiply by prefactor to start at the 'right'
point

# Calculate scale factor at t=18
a_now = a5[-1]

# Normalize scale factors
a1 /= a_now
a2 /= a_now
```

```
a3 /= a_now
a4 /= a_now
a5 /= a_now
a6 /= a_now


# Create the plot
fig, ax = plt.subplots(figsize=(12, 8))
ax.plot(np.log10(t1), np.log10(a1), color='r', label='Planck era',
lw=3)
ax.axvspan(np.log10(t1[0]), np.log10(t1[-1]), alpha=0.05, color='red')
ax.axvline(np.log10(t_i), color='b', ls='--', lw=1.5, alpha=0.4)
ax.axvspan(np.log10(t2[0]), np.log10(t2[-1]), alpha=0.05, color='b')
ax.plot(np.log10(t2), np.log10(a2), color='b', label='Inflation era',
lw=3)
ax.axvspan(np.log10(t3[0]), np.log10(t3[-1]), alpha=0.05, color='g')
ax.axvline(np.log10(t_f), color='g', ls='--', lw=1.5, alpha=0.4)
ax.plot(np.log10(t3), np.log10(a3), color='g', label='Radiation era',
lw=3)
ax.axvline(np.log10(t3[-1]), color='y', ls='--', lw=1.5, alpha=0.4)
ax.plot(np.log10(t4), np.log10(a4), color='y', label='Matter era',
lw=3)
ax.axvspan(np.log10(t4[0]), np.log10(t4[-1]), alpha=0.05, color='y')
ax.axvline(np.log10(t4[-1]), color='m', ls='--', lw=1.5, alpha=0.4)
ax.plot(np.log10(t5), np.log10(a5), color='m', label='Dark energy era',
lw=3)
ax.axvspan(np.log10(t5[0]), np.log10(t5[-1]), alpha=0.05, color='m')
ax.plot(np.log10(t6), np.log10(a6), color='k', label='Observable
horizon', lw=3, ls='--')
ax.tick_params(axis='both', which='both', direction='in', bottom=True,
top=True, left=True, right=True, labelsize=18)
plt.xlim(-44, 18)

plt.xlabel("Log(Time) [log(s)]", fontsize=18)
plt.ylabel("Log(Scale factor) [log(a)]", fontsize=18)
plt.title("Scale factor evolution in a five-phase universe",
fontsize=18)
plt.grid(alpha=0.2)
plt.legend(fontsize=18, fancybox=False, edgecolor='k', loc='lower
right')
plt.show()
```

Q3 (i) Random number generator

```
np.random.seed(42) # set seed for reproducibility
num_samples = 100
x = np.random.uniform(-1, 1, num_samples)

# plot results
plt.figure(figsize=(12, 5))
plt.axhline(0, color='lightsteelblue', linestyle='-', alpha=0.5)
plt.xlim(-1, 100)
plt.plot(x, color='cornflowerblue')
plt.xlabel('i-th iteration', fontsize=18)
plt.ylabel('overdensity, $\delta$', fontsize=18)
plt.show()
```

Q3 (ii) Random number generator + top-hat filters

```
w1 = 3 # set window size ie. the number of points to average over
mask1=np.ones((1,w1))/w1
mask1=mask1[0,:]

w2 = 15 # set window size
mask2=np.ones((1,w2))/w2
mask2=mask2[0,:]
print(f'3 Pt. Running Av. Mask: {mask1}')
print(f'15 Pt. Running Av. Mask: {mask2}')
convovled_3 = np.convolve(x, mask1, mode='same') # add padding of zeros
around the boundaries to preserve the length
convovled_15 = np.convolve(x, mask2, mode='same') # add padding of
zeros around the boundaries to preserve the length
# plot results
plt.figure(figsize=(12, 5))
plt.axhline(0, color='lightsteelblue', linestyle='-', alpha=0.5)
plt.plot(x, color='cornflowerblue', label='Original data')
plt.plot(convovled_3, label='3-Pt. MA', color='darkblue')
plt.plot(convovled_15, label='15-Pt. MA', color='r')
plt.legend()
plt.xlabel('i-th iteration', fontsize=18)
plt.xlim(-1, 100)
plt.ylabel('overdensity, $\delta$', fontsize=18)
plt.show()
```

Q3 (iii) Box Muller Transformation – Scatter Plot

```
num_samples = 10000 # use large num of samples to get a smooth distribution
u1, u2 = np.random.uniform(0, 1, num_samples), np.random.uniform(0, 1, num_samples) # sample uniform random numbers
z1, z2 = np.sqrt(-2*np.log(u1))*np.cos(2*np.pi*u2), np.sqrt(-2*np.log(u1))*np.sin(2*np.pi*u2) # apply transformation
fig, ax = plt.subplots(1, 2, figsize=(12, 5))
plt.axis('equal')
ax[0].scatter(u1, u2, s=1)
ax[0].set_xlabel('$u_1$', fontsize=18)
ax[0].set_ylabel('$u_2$', fontsize=18)
ax[0].set_title('Before Box-Muller', fontsize=15)
ax[1].scatter(z0, z1, s=1)
ax[1].set_xlabel('$z_1$', fontsize=18)
ax[1].set_ylabel('$z_2$', fontsize=18)
ax[1].set_title('After Box-Muller', fontsize=15)
plt.show()
```

Q3 (iii) Box Muller transformation - Histograms

```
def box_muller(u1, u2):
    z1 = np.sqrt(-2 * np.log(u1)) * np.cos(2 * np.pi * u2)
    z2 = np.sqrt(-2 * np.log(u1)) * np.sin(2 * np.pi * u2)
    return z1, z2
np.random.seed(0)
n = int(5E6) # use lots of samples to get a smooth histogram
u1 = np.random.rand(n)
u2 = np.random.rand(n)
bins = 1000
text_box_props = dict(boxstyle='round, pad=0.3', facecolor='white',
edgecolor='black')
z1, z2 = box_muller(u1, u2)
plt.figure(figsize=(15, 12))
plt.subplot(2, 2, 1)
plt.hist(u1, bins=bins, density=True, alpha=0.6, color='red')
plt.title("Before Box-Muller - u1", fontsize=20)
plt.xlabel("Value", fontsize=18)
plt.ylabel("Probability density", fontsize=18)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
mean = np.mean(u1)
std_dev = np.std(u1)
plt.text(0.95, 0.95, f"Mean: {mean:.4f}\nStd dev: {std_dev:.4f}",
         ha='right', va='top', transform=plt.gca().transAxes,
bbox=text_box_props, fontsize=18)
plt.subplot(2, 2, 2)
plt.hist(u2, bins=bins, density=True, alpha=0.6, color='orange')
plt.title("Before Box-Muller - u2", fontsize=20)
plt.xlabel("Value", fontsize=18)
plt.ylabel("Probability density", fontsize=18)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
mean = np.mean(u2)
std_dev = np.std(u2)
plt.text(0.95, 0.95, f"Mean: {mean:.4f}\nStd dev: {std_dev:.4f}",
         ha='right', va='top', transform=plt.gca().transAxes,
bbox=text_box_props, fontsize=18)
plt.subplot(2, 2, 3)
plt.hist(z1, bins=bins, density=True, alpha=0.6, color='red')
plt.title("After Box-Muller - z1", fontsize=20)
plt.xlabel("Value", fontsize=18)
plt.ylabel("Probability density", fontsize=18)
plt.xticks(fontsize=18)
```

```python
plt.yticks(fontsize=18)
plt.xlim(-6,6)
mean = np.mean(z1)
std_dev = np.std(z1)
plt.text(0.95, 0.95, f"Mean: {mean:.4f}\nStd dev: {std_dev:.4f}",
         ha='right', va='top', transform=plt.gca().transAxes,
bbox=text_box_props, fontsize=18)
plt.subplot(2, 2, 4)
plt.hist(z2, bins=bins, density=True, alpha=0.6, color='orange')
plt.title("After Box-Muller - z2", fontsize=20)
plt.xlabel("Value", fontsize=18)
plt.ylabel("Probability density", fontsize=18)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.xlim(-6,6)
mean = np.mean(z2)
std_dev = np.std(z2)
plt.text(0.95, 0.95, f"Mean: {mean:.4f}\nStd dev: {std_dev:.4f}",
         ha='right', va='top', transform=plt.gca().transAxes,
bbox=text_box_props, fontsize=18)
plt.tight_layout()
#plt.savefig("box_muller.png", dpi=800)
plt.show()
```

Q3 (iv) Gaussian convolution

```python
from scipy.ndimage import gaussian_filter1d
gaussian_filtered = gaussian_filter1d(x, sigma=1, mode='constant', cval=0.0) # for consistency, add padding of zeros around the boundaries to preserve the length

# plot results
plt.figure(figsize=(12, 5))
plt.axhline(0, color='k', linestyle='-', alpha=0.5)
plt.plot(x, label='Original data', alpha=0.5)
plt.plot(convovled_3, label='3-Pt. MA', alpha=0.5)
plt.plot(convovled_15, label='15-Pt. MA', color='r', alpha=0.5)
plt.plot(gaussian_filtered, label='Gaussian filtered', color='r', linestyle='--')
plt.legend()
plt.xlabel('i-th iteration', fontsize=18)
plt.xlim(-1, 100)
plt.ylabel('overdensity, $\delta$', fontsize=18)
plt.show()
```